



PATENT  
Attorney Docket No.: 16869P-107800US  
Client Ref. No.: 340300186US01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

NAOKO IKEGAYA *et al.*

Application No.: 10/789,771

Filed: February 27, 2004

For: SUPPORT SYSTEM FOR DATA  
MIGRATION

Customer No.: 20350

Examiner: Unassigned

Technology Center/Art Unit: 2175

Confirmation No.: 1911

**PETITION TO MAKE SPECIAL FOR  
NEW APPLICATION UNDER M.P.E.P.  
§ 708.02, VIII & 37 C.F.R. § 1.102(d)**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

This is a petition to make special the above-identified application under MPEP § 708.02, VIII & 37 C.F.R. § 1.102(d). The application has not received any examination by an Examiner.

(a) The Commissioner is authorized to charge the petition fee of \$130 under 37 C.F.R. § 1.17(i) and any other fees associated with this paper to Deposit Account 20-1430.

02/18/2005 AWONDAF1 00000041 201430 10789771  
01 FC:1464 130.00 DA

(b) All the claims are believed to be directed to a single invention. If the Office determines that all the claims presented are not obviously directed to a single invention, then Applicants will make an election without traverse as a prerequisite to the grant of special status.

(c) Pre-examination searches were made of U.S. issued patents, including a classification search, a computer database search, a keyword search, and a literature search. The searches were performed on or around November 10, 2004, and were conducted by a professional search firm, Kramer & Amado, P.C. The classification search covered Class 711 (subclasses 100 and 112) for the U.S. and foreign subclasses identified above. The computer database search was conducted on the USPTO systems EAST and WEST. The keyword search was conducted in Class 707 (subclasses 201, 202, 203, and 205), Class 709 (subclasses 203, 213, and 219), and Class 711 (subclasses 111, 114, 162, 163, and 165). The literature search was conducted on the Internet and commercial databases for relevant nonpatent documents. The inventors further provided two references considered most closely related to the subject matter of the present application (see references #5-6 below), which were cited in the Information Disclosure Statements filed on February 27, 2004.

(d) The following references, copies of which are attached herewith, are deemed most closely related to the subject matter encompassed by the claims:

- (1) U.S. Patent Publication No. 2003/0093439 A1;
- (2) U.S. Patent Publication No. 2004/0049553 A1;
- (3) U.S. Patent No. 6,598,174 B1;
- (4) U.S. Patent No. 6,571,354 B1;
- (5) U.S. Patent No. 6,108,748; and
- (6) Japanese Patent Publication No. JP 2000-187608. ✓

(e) Set forth below is a detailed discussion of references which points out with particularity how the claimed subject matter is distinguishable over the references.

A. Claimed Embodiments of the Present Invention

The claimed embodiments relate to a support system for data migration suitable for providing support for efficiently migrating data from an old storage device to a new storage device and removing the old storage device within a computer system, particularly in a large-scale computer system, to which a plurality of storage devices are connected.

Independent claim 1 recites a data migration support system, comprising a computer; a storage device connected to the computer; means for registering identification information about an unnecessary file to identify one or more unnecessary files; first means for determining whether a file stored on the storage device is one of the unnecessary files; and second means for determining, for a file that is not one of the unnecessary files, whether the file is to be migrated to another storage device.

Independent claim 9 recites a data migration support program for a computer system that is connected to a computer and a removable storage device or a storage device for storing data on a removable medium. The program comprises code for registering identification information about an unnecessary file to identify one or more unnecessary files; code for judging whether a file stored on the removable storage device or the removable storage medium is one of the unnecessary files; code for if the file stored on the removable storage device or the removable storage medium is not one of the unnecessary files, then judging whether the file is migrated to another storage device; and code for, if each file stored on the removable storage device or the removable storage medium has been judged to be one of the unnecessary files or has migrated to the another storage device, displaying information to indicate that data migration is completed for the removable storage device or the removable storage medium.

Independent claim 12 recites a data migration support device connected via a network to a first computer system that includes a removable storage device or a storage device for storing data on a removable storage medium and a second computer connected to the removable storage device or the storage device for storing data on a removable storage medium. The device comprises a processor; a controller; a memory; a communication device connected to the network; and a display unit. The processor registers identification information about an unnecessary file with the memory to identify a plurality of registered

unnecessary files, judges whether a file stored on the removable storage device or the removable storage medium is one of the registered unnecessary files, and if the file is not a registered unnecessary file, then judging whether the file is migrated to another storage device. If each file stored on the removable storage device or the removable storage medium has been determined to be one of the registered unnecessary files or has been migrated to the another storage device, then the controller outputs information to the display unit to indicate that data migration is completed for the removable storage device or the removable storage medium.

Independent claim 14 recites a method for identifying a removable storage device within a computer system that can be connected to a computer and a plurality of storage devices. The method comprises registering identification information about an unnecessary file and generating an unnecessary-file database; acquiring identification information about a file stored on the storage device; judging based on the unnecessary-file database whether the identification information about the file conforms to unnecessary-file conditions; judging whether the file is already migrated to another storage device; and if all the files stored on the storage device conform to the unnecessary-file conditions or are already migrated to the another storage device, then generating identification information about the storage device.

Independent claim 17 recites a method for identifying a storage medium that has completed data migration for a computer system to which a computer and a storage device for storing data on a removable storage medium are connected. The method comprises registering identification information about an unnecessary file and generating an unnecessary-file database; acquiring identification information about a file stored on the storage medium; judging with the unnecessary-file database whether the identification information about the file conforms to unnecessary-file conditions; judging whether a file corresponding to the nonconforming identification information about the file that is stored in the file database is already migrated to another storage device; and generating, if all files stored on the storage device conform to the unnecessary-file conditions or are already migrated to another storage device, identification information about the storage device.

Independent claim 20 recites a system, comprising a first computer; a storage device connected to the first computer; another storage device connected to the first computer; and a second computer connected via a network to the first computer and the

another storage device. The storage device acquires a history of file transfer to the another storage device. The second computer comprises a controller, a memory, a communication device connected to the network, and a display unit. The controller registers identification information about unnecessary files with the memory; judges whether files stored on the storage device coincides with the registered unnecessary files; acquires the transfer history from the storage device via the network and the communication device if the files stored on the storage device do not match the identification information about the unnecessary files, and judges whether the files stored on the storage device are migrated to another storage device; and, if each files stored on the storage device is the registered unnecessary file or file migrated to the another storage device, displays information on the display unit to indicate that data migration is completed for the removable storage device.

One of the benefits that may be derived is that it provides a method for providing support for data migration, identifying a removable storage device accurately and quickly, and identifying a storage medium that has completed data migration in a computer system to which a removable storage device or a storage device for storing data on a removable storage medium is connected.

B. Discussion of the References

1. U.S. Patent Publication No. 2003/0093439 A1

This reference relates to a method and apparatus for relocating data related to database management system in which a data storage position management main program collects volume physical storage position management information and physical storage apparatus operation information from a storage apparatus, and stores them as storage apparatus arrangement information and storage apparatus operation information. The program also collects necessary information from schema information in a DBMS, stores it as DBMS schema information, collects mapping information in a host and virtual volume information in a virtual volume switch, stores them as data storage position management information, acquires execution history information from the host, and stores it as a DBMS execution history information. See paragraphs [0060]-[0077].

The reference does not teach registering identification information about an unnecessary file to identify one or more unnecessary files; determining whether a file stored

on the storage device is one of the unnecessary files; and determining, for a file that is not one of the unnecessary files, whether the file is to be migrated to another storage device, as recited in independent claims 1, 9, 12, 14, 17, and 20.

2. U.S. Patent Publication No. 2004/0049553 A1

This reference discloses an information processing system for migrating data from a migration source storage subsystem in which a storage area has been housed from a host to a migration target storage subsystem. Configuration information is read out of the migration source storage subsystem, and on the basis of the information, the storage subsystem of the data migration target will be configured and a storage area will be provided.

The reference does not teach registering identification information about an unnecessary file to identify one or more unnecessary files; determining whether a file stored on the storage device is one of the unnecessary files; and determining, for a file that is not one of the unnecessary files, whether the file is to be migrated to another storage device, as recited in independent claims 1, 9, 12, 14, 17, and 20.

3. U.S. Patent No. 6,598,174 B1

This reference discloses a method and apparatus used in a storage network that facilitates the protection of data in, and replacement of, storage devices about to fail before the failure happens. The method provides for selecting a particular spare storage device, which can be used in place of the first particular storage device. In response to detecting the condition, data stored in the first particular storage device is migrated to the second particular storage device, and the second particular storage takes the place of the first particular storage device in the non-redundant array.

The reference does not teach registering identification information about an unnecessary file to identify one or more unnecessary files; determining whether a file stored on the storage device is one of the unnecessary files; and determining, for a file that is not one of the unnecessary files, whether the file is to be migrated to another storage device, as recited in independent claims 1, 9, 12, 14, 17, and 20.

4. U.S. Patent No. 6,571,354 B1

This reference relates to a method for managing data in a network, comprising, detecting a condition of a first particular storage device in a particular set of storage devices in a plurality of sets of storage devices; in response to detecting a condition, migrating the data set stored in the first particular storage device to the second particular storage device and identifying the second particular storage device as a member of the first particular set; and, determining whether a spare device is available for use in the first particular set of storage devices, and if a spare device is not available, then enabling the migrating of the data set to the second particular storage device.

The reference does not teach registering identification information about an unnecessary file to identify one or more unnecessary files; determining whether a file stored on the storage device is one of the unnecessary files; and determining, for a file that is not one of the unnecessary files, whether the file is to be migrated to another storage device, as recited in independent claims 1, 9, 12, 14, 17, and 20.

5. U.S. Patent No. 6,108,748

This reference relates to on-line, real-me, transparent data migration from an existing storage device to a replacement storage system. The existing and replacement storage devices are connected as a composite storage device that is coupled to a host network, or other data processing system. The replacement storage device includes a table which identifies data elements that have migrated to the replacement storage device. When a host system makes a data transfer request for one or more data elements, the replacement storage device determines whether the data elements have been migrated. If the data elements have migrated, the replacement storage device responds to the data transfer request independently of any interaction with the existing storage device. If the data elements have not migrated, the replacement storage device migrates the requested data elements and then responds to the data request and updates the data element map or table. When not busy servicing other requests, the replacement storage device operates in a background mode to migrate data elements so the data migration can occur concurrently with and transparently to system operations.

As discussed in the present application at page 3, line 20 to page 4, line 19, when a large-scale system is used so that data migration involves both a data owner and a system administrator, it is difficult to accurately determine whether data other than the selected migration data in the storage device to be removed is unnecessary using the method as disclosed in this reference. If the same data is duplicated in a large number of storage devices, it means that the storage devices are wasted unless such duplication aims at providing high-speed access or reliability enhancement. Even when the entire data stored on a certain storage device or storage medium is unnecessary or the same data is stored on another storage device, such storage device or storage medium may be left unremoved. Such results are contrary to the demand for effective use of storage devices and storage media.

The reference does not teach registering identification information about an unnecessary file to identify one or more unnecessary files; determining whether a file stored on the storage device is one of the unnecessary files; and determining, for a file that is not one of the unnecessary files, whether the file is to be migrated to another storage device, as recited in independent claims 1, 9, 12, 14, 17, and 20.

6. Japanese Patent Publication No. JP 2000-187608

This reference discloses a technique to execute data movement between new and old storage devices by file units without any interference of a host computer. A new storage device subsystem 3 is provided with two interfaces 31, 32, and connected with a host 1 and an old storage device subsystem 2. The new storage device subsystem 3 copies the data of a volume in the old storage device subsystem 2 to its own volume in the order of the head, and records the position at which the copy is completed in a copy pointer 43. The new storage device subsystem 3 accepts access from the host 1 as if it is access from the hold storage device subsystem 2.

As discussed in the present application at page 3, line 20 to page 4, line 19, when a large-scale system is used so that data migration involves both a data owner and a system administrator, it is difficult to accurately determine whether data other than the selected migration data in the storage device to be removed is unnecessary using the method as disclosed in this reference. If the same data is duplicated in a large number of storage devices, it means that the storage devices are wasted unless such duplication aims at



providing high-speed access or reliability enhancement. Even when the entire data stored on a certain storage device or storage medium is unnecessary or the same data is stored on another storage device, such storage device or storage medium may be left unremoved. Such results are contrary to the demand for effective use of storage devices and storage media.

The reference does not teach registering identification information about an unnecessary file to identify one or more unnecessary files; determining whether a file stored on the storage device is one of the unnecessary files; and determining, for a file that is not one of the unnecessary files, whether the file is to be migrated to another storage device, as recited in independent claims 1, 9, 12, 14, 17, and 20.

(f) In view of this petition, the Examiner is respectfully requested to issue a first Office Action at an early date.

Respectfully submitted,



Chun-Pok Leung  
Reg. No. 41,405

TOWNSEND and TOWNSEND and CREW LLP  
Two Embarcadero Center, 8<sup>th</sup> Floor  
San Francisco, California 94111-3834  
Tel: 650-326-2400  
Fax: 415-576-0300  
Attachments  
RL:rl  
60404459 v1

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-187608

(43)Date of publication of application : 04.07.2000

(51)Int.Cl.

G06F 12/00  
G06F 3/06

(21)Application number : 10-366346

(71)Applicant : HITACHI LTD

(22)Date of filing : 24.12.1998

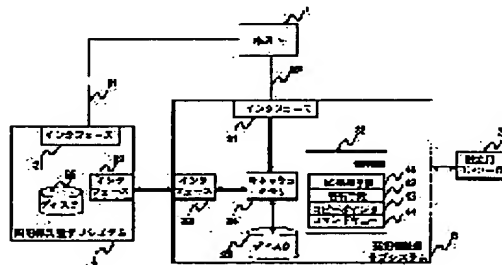
(72)Inventor : KITAMURA MANABU  
YAMAKAMI KENJI  
KURANO AKIRA  
KANBAYASHI KIMISAKI

### (54) STORAGE DEVICE SUB-SYSTEM

#### (57)Abstract:

PROBLEM TO BE SOLVED: To execute data movement between new and old storage devices by file units without any interference of a host computer.

SOLUTION: A new storage device sub-system 3 is provided with two interfaces 31 and 32, and connected with a host 1 and an old storage device sub-system 2. The new storage device sub-system 3 copies the data of a volume in the old storage device sub-system 2 to its own volume in the order of the head, and records the position at which the copy is completed in a copy pointer 43. Also, the new storage device sub-system 3 accepts access from the host 1 as if it is access from the old storage device sub-system 2.



### LEGAL STATUS

[Date of request for examination]

07.06.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than

the examiner's decision of rejection or  
application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's  
decision of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

\* NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

CLAIMS

---

[Claim(s)]

[Claim 1] It has the following and is said 1st [ the ]. A storage subsystem to said 2nd [ the ] A storage subsystem characterized by performing data shift to a storage subsystem, without stopping processing of said computer. The 1st A storage subsystem The 2nd A storage subsystem The 1st thru/or the 2nd With a computing system which consists of computers linked to a storage subsystem [ said ] Said 1st [ the ] A storage subsystem is said 2nd [ the ]. It connects with a storage subsystem and is said 2nd [ the ]. A storage subsystem Said 1st [ the ] It connects with a storage subsystem and said computer, and is (A). Said 1st [ the ] Data of a specific storage region of a storage subsystem It is said 2nd [ the ] to order from a head. A means to transmit to a storage region where a storage subsystem corresponds (B) A positional information storage means to record to which field a transfer was carried out by said transfer means (A) When a read-out demand from said computer is received, (C) From said positional information storage means It judges whether data is transfer ending to said 2nd storage subsystem. When finishing [ a transfer ], it is said 2nd [ the ]. Transmit data in a storage subsystem, and in not being transfer ending Said 1st [ the ] A storage region where a storage subsystem corresponds to said 2nd [ the ] Data is read to a storage subsystem. A means to transmit data to said computer, and (D) When a write request from said computer is received, it is said 2nd [ the ]. A storage subsystem and said 1st [ the ] A means which writes data in a storage region where a storage subsystem corresponds

[Claim 2] A storage subsystem characterized by transmitting preferentially a field where access frequency in said specific storage region is high in a storage subsystem according to claim 1.

[Claim 3] Setting to a storage subsystem according to claim 1, said specific storage region is said 1st [ the ]. Or the 2nd A storage subsystem characterized by being the file which a computer linked to a storage subsystem manages.

---

[Translation done.]

## \* NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] This invention relates to the storage system in information processing system etc., and relates to the computing system equipped with the function to perform the data shift between storage systems.

[0002]

[Description of the Prior Art] In the computer system used in a bank, a company, etc., since very a lot of data is dealt with, large-scale storage is needed. Furthermore, since these data increases every day, it will need to extend or transpose storage to large capacity storage. When replacing storage, as a usual means, it is storage (the henceforth, old storage). It is new storage (henceforth, new storage) about it for a magnetic tape etc. in backup. It realizes by restoring.

[0003] However, in order to take backup, it is once a host's I/O. It is necessary to stop and, generally the stop time is long. And the more the capacity of storage becomes large, the more this stop time increases. Furthermore, since operation non-stopped [ 24 hour ] is a premise in the above large systems and the same is said of the case of replacement of storage as for this, it is once I/O by the side of a host. The method which stops and takes backup has a problem.

[0004] By U.S. Pat. No. 5,680,640, in case data is shifted to new storage from the old storage, the method of performing on-line is shown. here -- the inside of new storage -- each address (truck) of the volume in the old storage every -- it has a table and memorizes for every truck whether shift of data was completed from the old storage to new storage. It is I/O from a host during shift. When there is a demand, it opts for actuation with reference to the table. For example, record which had the lead demand with reference to the table when there was a lead demand (block) It confirms whether have shifted to new storage, and when the data from the old storage has not shifted, data is read from the old storage. If data is in new storage, data will be read from new storage. Moreover, when there is a light demand, data writing is performed to new storage and a table is updated.

[0005]

[Problem(s) to be Solved by the Invention] By the method shown by U.S. Pat. No. 5,680,640, a table must be held about all trucks in the old storage. Usually, although a table uses that [-izing / semiconductor memory / with the battery back-up etc. / that / un-volatilizing ], semiconductor memory of not much the large capacity in the relation of cost is not carried in storage. If the data volume of the old storage is large, the table of new storage will also become large so much, and new storage will become what has a very high price.

[0006] Moreover, by this method, in order to write data only in new storage, when a failure occurs in one of storage and shift processing stops during shift in the writing from the host under data shift, both of the storage will be in the condition that data has conflict.

[0007]

[Means for Solving the Problem] In order to solve the above-mentioned technical problem, a computing system by this invention takes the following configuration. A computer system in this invention consists

of host computers connected to the 1st storage subsystem, the 2nd storage subsystem, and the 1st and 2nd storage subsystem. The 1st storage subsystem has an interface linked to a host computer, and the 2nd storage subsystem has an interface linked to the 1st storage subsystem and host computer. It connects with the 1st storage at first, and a host computer is I/O. Although processed, in case data shift is performed from the 1st storage subsystem to the 2nd storage subsystem, connection of a host computer and the 1st storage intersubsystem is intercepted, and a host computer is once connected with the 2nd storage subsystem. In a condition that do not separate a transmission line physically but connection is intercepted logically (it becomes off-line) is sufficient as this connection cutoff.

[0008] A host computer and the 1st Shift will be started, if connection of a storage subsystem is intercepted and connection with the 2nd storage subsystem is completed. Shift processing is the 2nd. A storage subsystem is the 1st. A lead demand is advanced to a storage subsystem and it is the 1st. A storage subsystem and the 2nd Data is copied through a transmission line which connects a storage subsystem. The 2nd There is a pointer which records how far a copy was completed in a storage subsystem, and an advance condition of data shift can be known.

[0009] I/O from a host computer under shift A demand is the 2nd. A storage subsystem receives. It is the 2nd when there is a lead demand from a host computer. For a storage subsystem, with reference to a copy pointer, data with a demand is already the 2nd. It checks whether it exists in a storage subsystem, and is the 2nd. The data is passed to a host computer when it exists in a storage subsystem. It is the 1st when it does not exist. Data is read from a storage subsystem and a host computer is passed. Moreover, it is the 2nd when there is a light demand. A storage subsystem is the 1st fundamentally. The 2nd Data is written in both storage subsystems. Thereby, it is I/O from a host computer during shift. Also when there is a demand, conflict does not take place to data in a storage subsystem, but shift can be continued. Moreover, it is data when a light demand is during shift from a host computer The 1st and the 2nd By carrying out duplex writing to a storage subsystem, even if abnormalities occur in one of storage subsystems, data is guaranteed.

[0010] With an above-mentioned method, it is the 1st during shift processing. It is I/O from a host to a field which has not copied data from a storage subsystem. It is the 1st when a demand comes. Since access to a storage subsystem occurs, Processing speed does not increase, therefore it is the 2nd. Counting of the access frequency is carried out about each disk which a storage subsystem has, access from a host is high to which field, or statistics are taken, and there is also a method of performing shift processing preferentially from a field where access frequency is high at the time of shift. Since a copy is completed at an early stage by this about a field where access frequency is high, it is I/O from a host. It is the 1st even if a demand comes. Access to a storage subsystem can be pressed down few.

[0011] Furthermore, in shift of data, only not only shift of an entire volume but a specific field and a further specific file shift. In that case, there is also a method of shifting only a field on which a specific file is recorded with reference to file management information within the 2nd storage subsystem.

[0012]

[Embodiment of the Invention] (1st operation gestalt) The example of a configuration of the computing system which applied this invention to drawing 1 is shown. A computer system consists of a host 1, an old storage subsystem 2, and a new storage subsystem 3. The old storage subsystem 2 consists of an interface 22 for connecting with the interface 21 for connecting with a host 1, and the new storage subsystem 3, and a disk 23. The new storage subsystem 3 consists of the interface 32 for connecting with the interface 31 for connecting with a host 1, and the old storage subsystem 2, a disk 33, cache memory 34, a control unit 35, and a console 36 for a setup. A disk 23 and a disk 33 may constitute a disk array from disks which may exist in each equipment, respectively, and exist. [ two or more ] Moreover, a disk 23 and a disk 33 may not necessarily be disks of the same format, for example, are SCSI (Small Computer System Interface). In the case of the disk treating fixed-length data format like a disk, the capacity of a disk 33 should be just larger than a disk 23. Moreover, in the case of the disk treating the data of CKD (count key data) format, the capacity of a disk 33 is larger than a disk 23, and format (3380/3390 format etc.) should be just in agreement with it.

[0013] Generally, it consists of a CPU and memory and the program which controls the new storage

subsystem 3 in memory exists, and although a control unit 35 is performed, it is omitted in drawing of this invention. As that which takes the lead in this invention with a control unit 35, it is I/O. The processing means 41, the shift means 42, the copy pointer 43, and the command queue 44 exist. I/O The processing means 41 processes by receiving the read/write demand from a host 1. Moreover, the shift means 42 performs migration processing of the data from the old storage subsystem 2 to the new storage subsystem 3. The copy pointer 43 is used with the shift means 42, and it records how far shift processing of data is progressing. The command queue 44 is I/O from a host 1. It is used in order to accept a demand. I/O from a host 1 A demand once enters here and is I/O. It is processed by the processing means 41. The console 36 for a setup changes various setup of the new storage subsystem 3. For example, it performs issuing the processing directions at the time of shifting data to the new storage subsystem 3 from the old storage subsystem 2 changed into the condition that a disk 33 is made into off-line and it can be used from a host 1 etc.

[0014] Drawing 2 explains the data shift processing to the new storage subsystem 3 from the old storage subsystem 2.

[0015] It connects with the host 1 in the transmission line 51, and the old storage subsystem 2 is performing I/O with a host in the beginning. In case data is shifted to the new storage subsystem 3 from the old storage subsystem 2, a host's 1 I/O is once stopped and connection of a transmission line 51 is cut. And it connects with the old storage subsystem 2 by # transmission 13#, and the new storage subsystem 3 is changed a host 1 into the condition of drawing 1 in a transmission line 52. In order that this activity may only change connection of the old storage subsystem 2 to connection between a host 1 and the new storage subsystem 3 with a host 1, only slight time amount should stop I/O from a host.

[0016] Shift processing can be started if it becomes the configuration of drawing 1. First, it is the copy pointer 43 in a control unit 35 0 It initializes (step 1001). It is confirmed whether, with reference to the inside of a command queue, the demand of the read/write processing from a host 1 is coming by step 1002 (step 1003). When there is a processing demand, processing in a queue is performed (step 1004). Although various methods, such as performing or performing only one of the processing of the altogether, can be considered when two or more processing demands exist, suppose that only one is performed here. Processing here is explained later.

[0017] At step 1005, the lead demand of the field which the copy pointer 43 shows to the old storage subsystem 2 is sent. That what is necessary is just the multiple of 1 sector if disks 23 and 33 are FBA format, if the unit of a lead is the disk of CKD format, it should just be the multiple of a truck. In this example, a disk makes a lead unit every sector in FBA format.

[0018] In addition, there may also be mounting which specifies a lead unit from the console 36 for a setup. Selection of enlarging a lead unit for this performing shift processing a little preferentially rather than the I/O process which comes from a host, or making a lead unit small in order to make delay hard to produce conversely at the usual I/O process is attained.

[0019] Reception is written out at step 1006 and data is written out from the old storage subsystem 2 for lead data to the address concerned of the new storage subsystem 3 at step 1007. It adds to the copy pointer 43 one time at step 1008. Whether the copy pointer 43 reached at step 1009 more than the maximum block count of a disk 23, and when checking and having not reached yet, return and processing are repeated to step 1002. Shift processing will be ended if the maximum block count is reached.

[0020] Next, it explains that processing with the new storage subsystem 3 when the lead demand from a host 1 comes by drawing 3 during shift processing flows. First, the lead place address by which it came from the host 1 is compared with the copy pointer 43 currently held with the new storage subsystem 3 (step 1101). Since data is already copied to the new storage subsystem 3 from the old storage subsystem 2 when the copy pointer 43 is larger than the lead place address, data can be read from the new storage subsystem 3 (step 1102), and lead processing can be performed by sending data to a host.

[0021] Conversely, since the contents of the old storage subsystem 2 are not reflected in the new storage subsystem 3 when the copy pointer 43 is a value below the lead place address at step 1101, lead (step 1105) processing is completed by sending to a host the data which received the lead demand from the

old storage subsystem 2 which receives lead data from delivery (step 1103) and the old storage subsystem 2 (step 1104) to the old storage subsystem 2.

[0022] Next, it explains that processing with the new storage subsystem 3 when the light demand from a host 1 comes by drawing 4 during shift processing flows. At step 1201, in order to write the light data first received from the host 1 in the old storage subsystem 2, a light demand is advanced to the old storage subsystem 2, and data is written in. Step 1202 compares the light point address by which it came from the host 1, and the copy pointer 43 currently held with the new storage subsystem 3. When the light point address is smaller than the copy pointer 43, the light data which came from the host 1 is written in a disk 33. Moreover, since a data copy is performed [ later ] by shift processing from the old storage subsystem 2 to the new storage subsystem 3 when the light point addresses are 43 or more copy pointers, it is not necessary to write in data to a disk 33 now, and light processing is ended, without doing anything to the new storage subsystem 3.

[0023] (2nd operation gestalt) The example of a configuration of the computing system which applied this invention to drawing 5 is shown. This example is also the 1st. The same configuration as an operation gestalt is taken and it consists of a host 1, an old storage subsystem 2, and a new storage subsystem 3. In this example, two or more disks 23 in the old storage subsystem 2 and the new storage subsystem 3 and disks 33 exist.

[0024] Generally, it consists of a CPU and memory and the program which controls the new storage subsystem 3 in memory exists, and although a control unit 35 is performed, it is omitted in drawing of this invention. As what is different from the 1st operation gestalt with a control unit 35, two or more copy pointers 143 exist, and two or more access counters 145 corresponding to the copy pointer 143 exist. Drawing 6 explains a copy pointer and an access counter. At this example, they are a copy pointer, an access counter, the start address of a field, and the last address 1 It has managed as a record of \*\*. An access counter 145 is I/O from a host 1 to the field of disk 33 throat. It is used in order to total whether there was any demand. This investigates much access to which field generally there is, and it is used for determining the sequence which begins data shift processing. For example, it is n from a head about each about two or more disks 33. I/O from a host [ as opposed to / divide into the field of an individual and / each field ] The count of a demand is recorded on an access counter 145. This looks at the orientation of much access to which field of a disk there is, and shift processing is preferentially carried out from the field considered that there is much access. The method which totals about all the disks 33 or is specified from the console 36 for a setup is sufficient as the disk which totals the count of access. The copy pointer 143 is also n. Individual preparation is carried out and it is matched with the access counter 145. Each copy pointer takes the value from the head of the field to last.

[0025] Drawing 7 explains the data shift processing to the new storage subsystem 3 from the old storage subsystem 2. It is the same as the 1st operation gestalt until it connects the new storage subsystem 3 with a host 1 before shift processing.

[0026] First, the copy pointer 143 in a control unit 35 is initialized (step 2001). In initialization, the peak value of two or more copy pointers is set as the start address of each field. Next, two or more copy pointers can be re(step 2002) ranked with order with high access frequency with reference to the access counter of each field. For example, the field of a disk 33 is trichotomized and it is drawing 6. When it is access frequency [ like ], it rearranges like drawing 8 . At step 2003, a shift means chooses a top copy pointer, i.e., the highest thing of access frequency. It is confirmed whether, with reference to the inside of the command queue 44, the demand of the read/write processing from a host 1 is coming by step 2004 (step 2005). When there is a processing demand, processing in a queue is performed (step 2006).

[0027] At step 2007, the lead demand of the field which a copy pointer shows to the old storage subsystem 2 is sent. Reception is written out at step 2008 and data is written out from the old storage subsystem 2 for lead data to the address concerned of the new storage subsystem 3 at step 2009. It adds to a copy pointer one time at step 2010. Whether the copy pointer 43 crossed the last address of a shift object domain at step 2011, and when checking and having not reached yet, return and processing are repeated to step 2004. If the maximum block count is reached, it will be confirmed whether there is any copy pointer processed next (step 2012). If the copy pointer which should be processed does not exist, it



ends, and when it exists, shift processing chooses as the following copy pointer, i.e., a degree, the pointer of the field where access frequency is high, and returns to step 2004.

[0028] Next, processing with the new storage subsystem 3 when the lead demand from a host 1 comes during shift processing is explained ( drawing 9 ).

[0029] First, the demand address by which it came from the host 1 confirms whether be the field which already carried out the completion of a copy (step 2101). The check method searches two or more copy pointers 143 from a head, and investigates whether it belongs to the range of which copy pointer. The copy pointer with which the demand address belongs understands whether the range of the copy pointer already carried out the completion of a copy only by judging the copy pointer front which is performing current shift processing, or the back. If it is the field which already carried out the completion of a copy, lead data will be read from the new storage subsystem 3 (step 2102), and data will be sent to a host (step 2105). When that is not right, lead data is received from delivery (step 2103) and the old disk unit subsystem 2 for a lead demand to the old storage subsystem 2 (step 2104).

[0030] Moreover, with the check in step 2101, when the demand address by which it came from the host 1 is in agreement with the field under current shift processing, it considers that the copy is not completed and step 2103 or subsequent ones is carried out. Or drawing 3 is processed, and the lead demand address may confirm whether it is equivalent to the completion portion of a copy in the field, and may choose processing.

[0031] Next, processing when a light demand comes by drawing 10 is explained. At step 2201, the demand address by which it came from the host 1 confirms whether be the field which already carried out the completion of a copy. Like the time of lead processing, the check method searches two or more copy pointers 143 from a head, and investigates to which range it belongs.

[0032] If it is the field which the copy already completed, the light data which came from the host 1 will be written in the disk 33 by the side of the new storage subsystem 3 (step 2202). Moreover, when the copy is not completed, in order to write the light data received from the host 1 in the old storage subsystem 2, a light demand is advanced to the old storage subsystem 2, and data is written in (step 2203).

[0033] Light processing is ended without the writing to the new storage subsystem 3 not writing in data to a disk 33 at this time, and carrying out any to the new storage subsystem 3, since a data copy is performed [ later ] by shift processing from the old storage subsystem 2 to the new storage subsystem 3. Or as it is in the 1st operation gestalt, the method of taking the art which writes in data before step 2201 at the old storage subsystem 2 side, and is changed into a condition [ writing / duplex ], and raising reliability may be used.

[0034] I/O [ as opposed to / to under shift processing / a non-copied field with the method of this example or the 1st operation gestalt ] If a demand comes, since access to the old storage subsystem 2 will occur, processing speed is affected. Therefore, about the high field of access, a processing speed fall is pressed down to I/O under shift processing by the method of this example which can finish data shift in an early phase.

[0035] (3rd operation gestalt) The example of a configuration of the computing system which applied this invention to drawing 11 is shown. A computer system consists of a host 1, an old storage subsystem 2, and a new storage subsystem 3, and the fundamental configuration is the same as the 1st operation gestalt.

[0036] As a different point from the 1st operation gestalt, it has the file analysis means 46 in the control unit 35 in the new storage subsystem 3. The write-in location of the specific file in the old storage subsystem 2 is recognized with the file analysis means 46, and it is 1. Data migration of only the file of \*\* is enabled. Although the file system which a host 1 uses, and the file system which can recognize the file analysis means 45 need to be in agreement with this, it shall be in agreement here. Moreover, the field information 147 for managing the location of a file exists. About this, it mentions later.

[0037] Drawing 12 explains the filing system on a disk unit. The filing system explained here is an example, and if the file analysis means 46 can recognize the other file system, it is applicable also in other file systems.

[0038] There is a volume label 71 in the specific location on a disk, and the address with which VTOC72 (Volume Table Of Contents) is written is describing at the volume label 71. As management information of the file currently written to the disk, a file name, the data format of a file, the positional information in which the file is stored are recorded on VTOC72. Each of these positional information is called DSCB here.

[0039] A file is put on the field called an extent. If an extent is the continuation field in which multiple tracks gathered and also has one case about one data set, it may exist more than plurality. When there are two or more extents, it is not necessary to exist in the field to which each extent continued. The data on a file is recorded sequentially from the head of the extent described at VTOC.

[0040] Drawing 13 explains the field information 147. The field information 147 holds the start address of the extent of a transitional object file, and the last address. When two or more extents exist, two or more information is held.

[0041] Drawing 14 and drawing 15 explain the data shift processing to the new storage subsystem 3 from the old storage subsystem 2. This processing serves as a portion (file analysis processing) which interprets the information on VTOC currently first written in the old storage subsystem 2 from two of the processings (data shift processing) which actually copy data. Processing initiation advances a demand from a host 1 or the console 36 for a setup to the new storage subsystem 3. The volume name in the old storage subsystem 2 used as a transitional object and a file name are specified in that case.

[0042] Drawing 14 explains file analysis processing. First, a lead demand is given to the old storage subsystem 2, a volume label is read, and it places on cache memory (step 3001). The read-out location of a volume label is realizable by reading the address of a certain regular location for a fixed position. A volume label is written in a disk 33 at step 3002. At step 3003, the location of VTOC is read from a volume label to ejection, and VTOC is read on cache memory. Next, reading and an object file are looked for for VTOC read on cache memory from a head, and the positional information is read (step 3004). At step 3005, an extent location is investigated from the positional information of the target file, and the head location of each extent and the last location are recorded on the field information 147. When there are two or more extents, two or more extent information is sorted, and it arranges from the smaller one of the address. The information on the object file of VTOC is written in a disk 33 at step 3006.

[0043] Drawing 15 explains shift processing. At step 3101, the copy pointer 43 in a control unit 35 is initialized to an extent start address. It is confirmed whether, with reference to the inside of a command queue, the demand of the read/write processing from a host 1 is coming by step 3102 (step 3103). When there is a processing demand, processing in a queue is performed (step 3104). At step 3105, the lead demand of the field which the copy pointer 43 shows to the old storage subsystem 2 is sent.

[0044] Reception is written out at step 3106 and data is written out from the old storage subsystem 2 for lead data to the address concerned of the new storage subsystem 3 at step 3107. It adds to the copy pointer 43 one time at step 3108. Whether the copy pointer 43 exceeded the extent last address at step 3109, and when checking and having not exceeded yet, return and processing are repeated to step 3102. If the maximum block count is reached, it will be confirmed whether there is any following extent with reference to the field information 147 (step 3110). If processing has finished about all extents, shift processing will be ended, and if there is the following extent, a copy pointer will be set as the following extent start address (step 3111).

[0045] Next, it explains that processing with the new storage subsystem 3 when the lead demand from a host 1 comes by drawing 16 during shift processing flows. First, the lead place address by which it came from the host 1 is compared with the information on the extent of the file for a copy currently held with the new storage subsystem 3, and it is confirmed whether, as for close, the lead place address is in the extent of the field information 147 (step 3201). When close is in an extent, the copy pointer 43 is compared with the lead place address below (step 3202). Since data is already copied to the new storage subsystem 3 from the old storage subsystem 2 when the copy pointer 43 is larger than the lead place address, data can be read from the new storage subsystem 3 (step 3203), and lead processing can be performed by sending data to a host.

[0046] When close is not in step 3201 extent, since the lead demand from a host is not access to the object file of this shift processing, it will read data from the old storage subsystem 2. Lead data is received from delivery (step 3204) and the old storage subsystem 2 for a lead demand to the old storage subsystem 2 (step 3205). Or also when the copy pointer 43 is judged at step 3202 to be below the lead place address, data will be read from the old storage subsystem 2. Lead (step 3206) processing is completed by finally sending data to a host.

[0047] Here, when many extents exist, the overhead of check processing of whether the lead place address in a step is in an extent starts. Therefore, mounting of in lead processing it not being concerned with the value of the lead place address, but carrying out and carrying out read-out from the old storage subsystem 2 unconditionally is sufficient.

[0048] Next, it explains that processing with the new storage subsystem 3 when the light demand from a host 1 comes by drawing 17 during shift processing flows. It is confirmed whether, as for close, at step 3301, the light point address is in the extent of the field information 147 by comparing first the light point address by which it came from the host 1 with the information on the extent of the file for a copy currently held with the new storage subsystem 3. Since it is the writing to the file set as the object of data shift when close is in an extent, a light demand is advanced to the old storage subsystem 2, and the light data received from the host 1 is written in (step 3302). Next, the copy pointer 43 is compared with the light point address (step 3303). Since a data copy is performed [ later ] by shift processing from the old storage subsystem 2 to the new storage subsystem 3 when the copy pointer 43 is smaller than the light point address, it is not necessary to write in data to a disk 33 now, and light processing is ended, without doing anything to the new storage subsystem 3. When that is not right, or when the light point address is judged at step 3301 to be the outside of an object extent, data is written in the disk 33 in the new storage subsystem 3 (step 3304).

[0049]

[Effect of the Invention] The data shift to a new storage subsystem from the old storage subsystem is possible, receiving the processing from a host computer in the computer system of this invention. Moreover, shift in a file unit is possible and only data to shift can be shifted alternatively. Moreover, since duplex writing of the data under shift is carried out to two, the old storage subsystem and a new storage subsystem, even if a failure occurs in one of storage subsystems during shift, it can continue shift processing.

---

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-187608

(P2000-187608A)

(43) 公開日 平成12年7月4日(2000.7.4)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
G 0 6 F 12/00	5 4 5	G 0 6 F 12/00	5 4 5 M 5 B 0 6 5
3/06	3 0 1	3/06	3 0 1 G 5 B 0 8 2

審査請求 未請求 請求項の数 3 O L (全 17 頁)

(21) 出願番号 特願平10-366346

(22) 出願日 平成10年12月24日(1998. 12. 24)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 北村 学

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 山神 憲司

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(74) 代理人 100068504

弁理士 小川 勝男

最終頁に続く

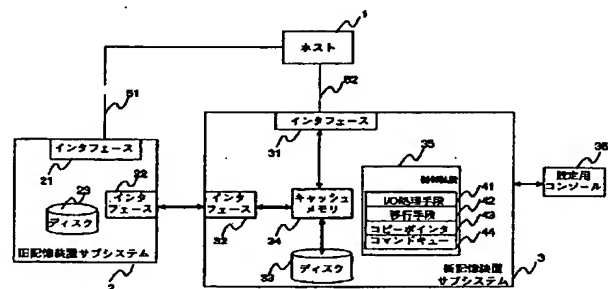
(54) 【発明の名称】 記憶装置サブシステム

(57) 【要約】

【課題】新旧の記憶装置間でのデータ移動をファイル単位でホスト計算機の介在なしで実施する。

【解決手段】新記憶装置サブシステム3は2つのインタフェース31、32を有し、ホスト1と旧記憶装置サブシステム2に接続されている。新記憶装置サブシステム3は、旧記憶装置サブシステム2内ボリュームのデータを、先頭から順に自ボリュームにコピーし、コピーが完了している位置をコピーポインタ43に記録する。また、その間、ホスト1からのアクセスを、あたかも旧記憶装置サブシステム2であるかのように受け付ける。

図 1



## 【特許請求の範囲】

【請求項 1】第 1 の記憶装置サブシステムと、第 2 の記憶装置サブシステムと、前記第 1 ないし第 2 の記憶装置サブシステムに接続する計算機とで構成される計算機システムで、前記第 1 の記憶装置サブシステムは前記第 2 の記憶装置サブシステムに接続され、また前記第 2 の記憶装置サブシステムは、前記第 1 の記憶装置サブシステムと前記計算機に接続され、

(A) 前記第 1 の記憶装置サブシステムの特定の記憶領域のデータを、先頭から順に前記第 2 の記憶装置サブシステムの対応する記憶領域に転送する手段

(B) 前記転送手段 (A) によってどの領域まで転送が実施されたかを記録する位置情報記憶手段と、

(C) 前記計算機からの読み出し要求を受け付けると、前記位置情報記憶手段から、前記第 2 の記憶装置サブシステムにデータが転送済みであるか否かを判定し、転送済みの場合には前記第 2 の記憶装置サブシステム内のデータを転送し、転送済みでない場合には、前記第 1 の記憶装置サブシステムの対応する記憶領域から前記第 2 の記憶装置サブシステムへとデータを読み出して、前記計算機へデータを転送する手段と、

(D) 前記計算機からの書き込み要求を受け付けると、前記第 2 の記憶装置サブシステムと前記第 1 の記憶装置サブシステムの対応する記憶領域とにデータを書き込む手段とを有し、前記第 1 の記憶装置サブシステムから前記第 2 の記憶装置サブシステムへのデータ移行を、前記計算機の処理を止めることなく実行することを特徴とする記憶装置サブシステム。

【請求項 2】請求項 1 記載の記憶装置サブシステムにおいて、前記特定の記憶領域内のアクセス頻度の高い領域を優先的に転送することを特徴とする記憶装置サブシステム。

【請求項 3】請求項 1 記載の記憶装置サブシステムにおいて、前記特定の記憶領域は前記第 1 ないし第 2 の記憶装置サブシステムに接続する計算機の管理するファイルであることを特徴とする記憶装置サブシステム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、情報処理システムなどにおける記憶装置システムに係り、記憶装置システム間のデータ移行を行う機能を備えた計算機システムに関する。

## 【0002】

【従来の技術】銀行や企業などで使用される計算機システムでは、極めて大量のデータが取り扱われるため、大規模な記憶装置を必要とする。さらに、それらデータは日々増加していくため、記憶装置を増設する、またはより大容量な記憶装置へ置き換えていく必要が生じる。記憶装置を置き換える場合、通常的手段としては、記憶装置(以下、旧記憶装置)のバックアップを磁気テープな

どにとり、それを新しい記憶装置(以下、新記憶装置)にリストアすることで実現する。

【0003】ただし、バックアップをとるためには一旦ホストの I/O を停止する必要がある、その停止時間は一般に長い。そしてこの停止時間は、記憶装置の容量が大きくなればなるほど増大していく。さらに、上記のような大規模システムでは 24 時間無停止運転が前提であり、これは記憶装置の置き換えの場合も同様であるので、一旦ホスト側の I/O を停止してバックアップをとる方式は問題がある。

【0004】米国特許 5,680,640 号では、旧記憶装置から新記憶装置へとデータを移行する際にオンラインで実行する方法が示されている。ここでは、新記憶装置内に、旧記憶装置内ボリュームの各アドレス(トラック)毎にテーブルを有し、旧記憶装置から新記憶装置へデータの移行が完了したかどうかをトラック毎に記憶する。移行中にホストからの I/O 要求があった場合には、そのテーブルを参照して動作を決定する。例えばリード要求があった場合、テーブルを参照して、そのリード要求のあったレコード(ブロック)が新記憶装置に移行されているかどうかチェックし、旧記憶装置からのデータが移行されていない時には旧記憶装置からデータを読み込む。新記憶装置内にデータがあれば新記憶装置からデータを読み出す。また、ライト要求があった場合には、新記憶装置にデータ書き込みを行い、テーブルの更新を行う。

## 【0005】

【発明が解決しようとする課題】米国特許 5,680,640 号で示されている方法では、旧記憶装置内全てのトラックについてテーブルを保持しなければならない。通常テーブルは半導体メモリをバッテリバックアップなどにより不揮発化したものを使用するが、コストの関係であり大容量の半導体メモリを記憶装置に搭載しない。旧記憶装置のデータ容量が大きいと、それだけ新記憶装置のテーブルも大きくなり、新記憶装置は非常に価格の高いものになってしまう。

【0006】また、この方法では、データ移行中のホストからの書き込みで、新記憶装置にのみデータを書き込むため、移行中にどちらかの記憶装置に障害が発生して移行処理が止まった場合、どちらの記憶装置もデータに矛盾のある状態となってしまう。

## 【0007】

【課題を解決するための手段】上記課題を解決するため、本発明による計算機システムは下記の構成をとる。本発明における計算機システムは第 1 の記憶装置サブシステムと、第 2 の記憶装置サブシステムと、第 1、第 2 の記憶装置サブシステムに接続されるホストコンピュータとで構成される。第 1 の記憶装置サブシステムはホストコンピュータに接続するインタフェースを有し、第 2 の記憶装置サブシステムは第 1 の記憶装置サブシステム

とホストコンピュータに接続するインタフェースとを有する。ホストコンピュータは、最初は第1の記憶に接続されてI/O処理を行うが、第1の記憶装置サブシステムから第2の記憶装置サブシステムへとデータ移行を行う際に、一旦ホストコンピュータと第1の記憶装置サブシステム間の接続は遮断され、ホストコンピュータは第2の記憶装置サブシステムと接続される。この接続遮断は物理的に伝送路を切り離すのではなく、論理的に接続が遮断される（オフラインになる）状態でも良い。

【0008】ホストコンピュータと第1の記憶装置サブシステムの接続が遮断され、第2の記憶装置サブシステムとの接続が完了すると、移行を開始する。移行処理は、第2の記憶装置サブシステムが第1の記憶装置サブシステムに対してリード要求を出し、第1の記憶装置サブシステムと第2の記憶装置サブシステムとを接続する伝送路を介してデータをコピーする。第2の記憶装置サブシステム内にはコピーがどこまで完了したかを記録するポイントがあり、データ移行の進行状況を知ることができる。

【0009】移行中のホストコンピュータからのI/O要求は、第2の記憶装置サブシステムが受け付ける。ホストコンピュータからリード要求があった場合、第2の記憶装置サブシステムはコピーポイントを参照し、要求のあったデータがすでに第2の記憶装置サブシステムに存在するかどうかを確認し、第2の記憶装置サブシステム内に存在する場合は、そのデータをホストコンピュータに渡す。存在しない場合には第1の記憶装置サブシステムからデータを読み出してホストコンピュータに渡す。また、ライト要求があった場合、第2の記憶装置サブシステムは基本的に、第1と第2の記憶装置サブシステムの両方にデータを書き込む。これにより、移行中にホストコンピュータからのI/O要求があった場合にも記憶装置サブシステム内のデータに矛盾は起こらず、移行を続行できる。また、移行中にホストコンピュータからライト要求があった場合には、データを第1、第2の記憶装置サブシステムに二重書きすることにより、どちらかの記憶装置サブシステムに異常が発生してもデータが保証される。

【0010】上記方法では、移行処理中に第1の記憶装置サブシステムからデータをコピーしていない領域に対してホストからI/O要求が来ると、第1の記憶装置サブシステムへのアクセスが発生するため、処理速度が上がらない、そのため、第2の記憶装置サブシステムの有する各ディスクについてアクセス頻度を計数し、どの領域にホストからのアクセスが高いか統計をとり、移行時には、アクセス頻度の高い領域から優先的に移行処理を実行する方法もある。これにより、アクセス頻度の高い領域についてコピーが早期に完了するため、ホストからのI/O要求が来ても第1の記憶装置サブシステムへのアクセスを少なく押えることができる。

【0011】さらに、データの移行ではボリューム全体の移行だけでなく、特定の領域、さらには特定のファイルだけ移行できれば良い場合もある。その場合には、第2の記憶装置サブシステム内で、ファイル管理情報を参照して、特定のファイルの記録されている領域のみを移行する方法もある。

【0012】

【発明の実施の形態】（第1の実施形態）図1に本発明を適用した計算機システムの構成例を示す。計算機システムは、ホスト1、旧記憶装置サブシステム2、新記憶装置サブシステム3から構成される。旧記憶装置サブシステム2はホスト1と接続するためのインタフェース21、新記憶装置サブシステム3と接続するためのインタフェース22、ディスク23で構成される。新記憶装置サブシステム3は、ホスト1と接続するためのインタフェース31、旧記憶装置サブシステム2と接続するためのインタフェース32、ディスク33、キャッシュメモリ34、制御装置35、設定用コンソール36で構成される。ディスク23、ディスク33はそれぞれ各装置内に複数存在しても良く、また複数存在するディスク同士でディスクアレイを構成してもよい。また、ディスク23、ディスク33は必ずしも同じ形式のディスクでなくても良く、例えばSCSI (Small Computer System Interface) ディスクのような固定長データ形式を扱うディスクの場合には、ディスク33の容量がディスク23よりも大きければ良い。また、CKD (カウントキーデータ) 形式のデータを扱うディスクの場合には、ディスク33の容量がディスク23よりも大きく、かつ、形式(3380/3390形式など)が一致していれば良い。

【0013】制御装置35は、一般的にはCPUとメモリで構成されて、メモリ中に新記憶装置サブシステム3を制御するプログラムが存在し、実行されるが、本発明の図では省略している。制御装置35で本発明の中心となるものとしては、I/O処理手段41、移行手段42、コピーポイント43、コマンドキュー44が存在する。I/O処理手段41はホスト1からのリード/ライト要求を受け、処理を行う。また移行手段42は旧記憶装置サブシステム2から新記憶装置サブシステム3へのデータの移動処理を行う。コピーポイント43は移行手段42で用いられ、データの移行処理がどこまで進んでいるかを記録しておく。コマンドキュー44はホスト1からのI/O要求を入れるために用いられる。ホスト1からのI/O要求は一旦ここに入り、I/O処理手段41によって処理される。設定用コンソール36は新記憶装置サブシステム3の各種設定の変更を行う。例えば、ディスク33をオフラインにしてホスト1から使用できる状態にする、旧記憶装置サブシステム2から新記憶装置サブシステム3へデータを移行する際の処理指示を出すなどを行う。

【0014】図2で、旧記憶装置サブシステム2から新

記憶装置サブシステム 3 へのデータ移行処理について説明する。

【0015】はじめは旧記憶装置サブシステム 2 は伝送路 51 でホスト 1 に接続されており、ホストとの I/O を行っている。旧記憶装置サブシステム 2 から新記憶装置サブシステム 3 へデータを移行する際に、一旦ホスト 1 の I/O を止めて伝送路 51 の接続を切る。そして新記憶装置サブシステム 3 は伝送路 52 でホスト 1 に、# 伝送 13 # で旧記憶装置サブシステム 2 と接続し、図 1 の状態にする。この作業はホスト 1 と旧記憶装置サブシステム 2 の接続をホスト 1 と新記憶装置サブシステム 3 との接続に切り替えるだけであるため、わずかな時間だけホストからの I/O を止めるだけで良い。

【0016】図 1 の構成になると移行処理を開始できる。まず、制御装置 35 内のコピーポインタ 43 を 0 に初期化する (ステップ 1001)。ステップ 1002 でコマンドキュー内を参照し、ホスト 1 からのリード/ライト処理の要求が来ているかチェックする (ステップ 1003)。処理要求がある場合には、キュー内の処理を実行する (ステップ 1004)。処理要求が複数個存在する場合、全てその処理を実行する、または 1 つだけ実行する、など、さまざまな方式が考えられるが、ここでは 1 つだけを実行することとする。ここでの処理については、後で説明する。

【0017】ステップ 1005 では、旧記憶装置サブシステム 2 にコピーポインタ 43 の示す領域のリード要求を送る。リードの単位は、ディスク 23、33 が FBA 形式であれば、1 セクタの倍数であれば良く、CKD 形式のディスクであればトラックの倍数であれば良い。本実施例では、ディスクは FBA 形式で、リード単位は 1 セクタごととする。

【0018】なお、リード単位を、設定用コンソール 36 から指定する実装もありうる。これにより、ホストから来る I/O 処理よりも移行処理をやや優先的に行いたい場合にはリード単位を大きくしたり、あるいは逆に通常の I/O 処理に遅れを生じにくくするためにリード単位を小さくするという選択が可能となる。

【0019】ステップ 1006 で旧記憶装置サブシステム 2 からリードデータを受け取り、ステップ 1007 で新記憶装置サブシステム 3 の当該アドレスにデータを書き出す。ステップ 1008 でコピーポインタ 43 に 1 加算する。ステップ 1009 でコピーポインタ 43 がディスク 23 の最大ブロック数以上に達したかチェックし、まだ達していない場合にはステップ 1002 に戻り、処理を繰り返す。最大ブロック数に達していれば、移行処理は終了する。

【0020】次に図 3 で、移行処理中にホスト 1 からのリード要求が来た場合の新記憶装置サブシステム 3 での処理の流れについて説明する。まず、ホスト 1 から来たリード先アドレスと、新記憶装置サブシステム 3 で保持

しているコピーポインタ 43 とを比較する (ステップ 1101)。コピーポインタ 43 がリード先アドレスよりも大きい場合には、新記憶装置サブシステム 3 には旧記憶装置サブシステム 2 からすでにデータがコピーされているので、新記憶装置サブシステム 3 からデータを読み出し (ステップ 1102)、ホストにデータを送ることでリード処理を行うことができる。

【0021】逆にステップ 1101 でコピーポインタ 43 がリード先アドレス以下の値の場合には、新記憶装置サブシステム 3 には旧記憶装置サブシステム 2 の内容が反映されていないため、旧記憶装置サブシステム 2 にリード要求を送り (ステップ 1103)、旧記憶装置サブシステム 2 からリードデータを受け取る (ステップ 1104) 旧記憶装置サブシステム 2 から受け取ったデータをホストに送ることで (ステップ 1105) リード処理が完了する。

【0022】次に図 4 で、移行処理中にホスト 1 からのライト要求が来た場合の新記憶装置サブシステム 3 での処理の流れについて説明する。ステップ 1201 で、まずホスト 1 から受け取ったライトデータを旧記憶装置サブシステム 2 に書き込むため、旧記憶装置サブシステム 2 に対してライト要求を出し、データを書き込む。ステップ 1202 で、ホスト 1 から来たライト先アドレスと、新記憶装置サブシステム 3 で保持しているコピーポインタ 43 とを比較する。ライト先アドレスがコピーポインタ 43 よりも小さい場合には、ディスク 33 にホスト 1 から来たライトデータを書き込む。また、ライト先アドレスがコピーポインタ 43 以上の場合には、後で移行処理によって旧記憶装置サブシステム 2 から新記憶装置サブシステム 3 へとデータコピーが行われるため、今ディスク 33 に対してデータを書き込む必要はなく、新記憶装置サブシステム 3 に対しては何もせずにライト処理は終了する。

【0023】(第 2 の実施形態) 図 5 に本発明を適用した計算機システムの構成例を示す。本例も第 1 の実施形態と同じ構成をとり、ホスト 1、旧記憶装置サブシステム 2、新記憶装置サブシステム 3 から構成される。本例では、旧記憶装置サブシステム 2、新記憶装置サブシステム 3 内のディスク 23、ディスク 33 は複数存在する。

【0024】制御装置 35 は、一般的には CPU とメモリで構成されて、メモリ中に新記憶装置サブシステム 3 を制御するプログラムが存在し、実行されるが、本発明の図では省略している。制御装置 35 で第 1 の実施形態と異なるものとして、コピーポインタ 143 は複数存在し、コピーポインタ 143 に対応した複数のアクセスカウンタ 145 が存在する。図 6 でコピーポインタとアクセスカウンタについて説明する。本例では、コピーポインタ、アクセスカウンタ、領域の先頭アドレス、最終アドレスを 1 つのレコードとして管理している。アクセス



カウンタ145は、ディスク33のどの領域にホスト1からのI/O 要求があったかを集計するために使用する。これにより、一般にどの領域へのアクセスが多いか調べ、データ移行処理を始める順番を決定するのに使用される。例えば、複数のディスク33について、それぞれを先頭からn 個の領域に分割し、各領域に対するホストからのI/O 要求回数をアクセスカウンタ145に記録する。これにより、ディスクのどの領域に対するアクセスが多いかの傾向を見て、アクセスの多いと考えられる領域から優先的に移行処理を実施する。アクセス回数を集計するディスクは、全てのディスク33について集計する、または設定用コンソール36から指定する方式でも良い。コピーポインタ143もn 個用意され、アクセスカウンタ145と対応付けられている。各コピーポインタはその領域の先頭から最終までの値をとる。

【0025】図7で、旧記憶装置サブシステム2から新記憶装置サブシステム3へのデータ移行処理について説明する。移行処理の前に、ホスト1と新記憶装置サブシステム3を接続するまでは第1の実施形態と同じである。

【0026】まず、制御装置35内のコピーポインタ143を初期化する(ステップ2001)。初期化では複数のコピーポインタの先頭値を各領域の先頭アドレスに設定する。次に、各領域のアクセスカウンタを参照し、複数のコピーポインタをアクセス頻度の高い順にならべかえる(ステップ2002)。例えば、ディスク33の領域を3分割して、図6のようなアクセス頻度であった場合、図8のように並べかえる。ステップ2003で、移行手段は先頭のコピーポインタ、すなわちアクセス頻度の最も高いものを選択する。ステップ2004でコマンドキュー44内を参照し、ホスト1からのリード/ライト処理の要求が来ているかチェックする(ステップ2005)。処理要求がある場合には、キュー内の処理を実行する(ステップ2006)。

【0027】ステップ2007では、旧記憶装置サブシステム2にコピーポインタの示す領域のリード要求を送る。ステップ2008で旧記憶装置サブシステム2からリードデータを受け取り、ステップ2009で新記憶装置サブシステム3の当該アドレスにデータを書き出す。ステップ2010でコピーポインタに1加算する。ステップ2011でコピーポインタ43が移行対象領域の最終アドレスを越えたかチェックし、まだ達していない場合にはステップ2004に戻り、処理を繰り返す。最大ブロック数に達していれば、次に処理するコピーポインタがあるかチェックする(ステップ2012)。処理すべきコピーポインタが存在しなければ移行処理は終了し、存在する場合には、次のコピーポインタ、すなわち次にアクセス頻度の高い領域のポインタを選択し、ステップ2004に戻る。

【0028】次に移行処理中にホスト1からのリード要

求が来た場合の新記憶装置サブシステム3での処理を説明する(図9)。

【0029】まず、ホスト1から来た要求アドレスが、すでにコピー完了した領域であるかどうかチェックする(ステップ2101)。チェック方法は、複数のコピーポインタ143を先頭からサーチし、どのコピーポインタの範囲に属するかを調べる。そのコピーポインタの範囲がすでにコピー完了したかどうかは、要求アドレスの属するコピーポインタが、現在移行処理を行っているコピーポインタの前か後かを判定するだけで分かる。すでにコピー完了した領域であれば、新記憶装置サブシステム3からリードデータを読み出し(ステップ2102)、ホストにデータを送る(ステップ2105)。そうでない場合には旧記憶装置サブシステム2にリード要求を送り(ステップ2103)、旧ディスク装置サブシステム2からリードデータを受け取る(ステップ2104)。

【0030】また、ステップ2101でのチェックで、ホスト1から来た要求アドレスが現在移行処理中の領域に一致した場合には、コピーが完了していないとみなしてステップ2103以降を実施する。あるいは図3の処理を実施して、リード要求アドレスが、その領域内のコピー完了部分に相当するかどうかをチェックして、処理を選択しても良い。

【0031】次に図10でライト要求が来た時の処理を説明する。ステップ2201で、ホスト1から来た要求アドレスが、すでにコピー完了した領域であるかどうかチェックする。チェック方法は、リード処理の時と同様、複数のコピーポインタ143を先頭からサーチし、どの範囲に属するかを調べる。

【0032】すでにコピーが完了した領域であれば、新記憶装置サブシステム3側のディスク33にホスト1から来たライトデータを書き込む(ステップ2202)。また、コピーが完了していない場合には、ホスト1から受け取ったライトデータを旧記憶装置サブシステム2に書き込むため、旧記憶装置サブシステム2に対してライト要求を出し、データを書き込む(ステップ2203)。

【0033】新記憶装置サブシステム3への書き込みは、後で移行処理によって旧記憶装置サブシステム2から新記憶装置サブシステム3へとデータコピーが行われるため、この時点でディスク33に対してデータを書き込む必要はなく、新記憶装置サブシステム3に対しては何もせずにライト処理は終了する。あるいは第1の実施形態にあるように、ステップ2201の前に旧記憶装置サブシステム2側にデータを書き込んで二重書きの状態にする処理方法をとって信頼性を高める方法でも良い。

【0034】本実施例あるいは第1の実施形態の方式では、移行処理中に未コピー領域に対するI/O 要求が来ると、旧記憶装置サブシステム2に対するアクセスが発



生するため、処理速度に影響が出る。そのため、アクセスの高い領域については早い段階でデータ移行を済ませることができる本例の方式では、移行処理中のI/Oに対して処理速度低下が押えられる。

【0035】(第3の実施形態)図11に本発明を適用した計算機システムの構成例を示す。計算機システムは、ホスト1、旧記憶装置サブシステム2、新記憶装置サブシステム3から構成され、基本的構成は第1の実施形態と同じである。

【0036】第1の実施形態と異なる点として、新記憶装置サブシステム3内の制御装置35にファイル解析手段46を有する。ファイル解析手段46によって旧記憶装置サブシステム2内の特定のファイルの書き込み位置を認識し、1つのファイルのみのデータ移動を可能にする。これにはホスト1の使用するファイルシステムと、ファイル解析手段45が認識できるファイルシステムが一致している必要があるが、ここではそれが一致しているものとする。また、ファイルの位置を管理するための領域情報147が存在する。これについては後述する。

【0037】図12で、ディスク装置上のファイル管理方式について説明する。ここで説明するファイル管理方式は一例であり、ファイル解析手段46がそれ以外のファイルシステムを認識できれば、他のファイルシステムにおいても適用可能である。

【0038】ディスク上の特定位置にはボリュームラベル71があり、ボリュームラベル71には、VTOC72 (Volume Table Of Contents) の書かれているアドレスが記されている。VTOC72にはそのディスクに書かれているファイルの管理情報として、ファイル名、ファイルのデータ形式、ファイルの格納されている位置情報などが記録される。この位置情報の1つ1つをここではDSCBと呼ぶ。

【0039】ファイルはエクステントと呼ばれる領域に置かれる。エクステントは複数トラックが集まった連続領域で、1つのデータセットにつき、1つの場合もあるが、複数以上存在することもある。エクステントが複数ある場合、各エクステントが連続した領域に存在している必要はない。ファイル上のデータは、VTOCに記されたエクステントの先頭から順に記録される。

【0040】図13で領域情報147について説明する。領域情報147は移行対象ファイルのエクステントの先頭アドレス、最終アドレスを保持する。エクステントが複数存在する時には複数の情報を保持する。

【0041】図14、図15で、旧記憶装置サブシステム2から新記憶装置サブシステム3へのデータ移行処理について説明する。この処理は、まず旧記憶装置サブシステム2内に書かれているVTOCの情報を解釈する部分(ファイル解析処理)と、実際にデータをコピーする処理(データ移行処理)の2つからなる。処理開始はホスト1または設定用コンソール36から、新記憶装置サ

ブシステム3に対して要求を出す。その際、移行対象となる旧記憶装置サブシステム2内ボリューム名とファイル名を指定する。

【0042】図14でファイル解析処理を説明する。まず、旧記憶装置サブシステム2にリード要求を出し、ボリュームラベルを読み出してキャッシュメモリ上に置く(ステップ3001)。ボリュームラベルの読み出し位置は固定位置のため、ある決まった位置のアドレスを読み出すことにより実現できる。ステップ3002でボリュームラベルをディスク33に書き込む。ステップ3003ではボリュームラベルからVTOCの位置を取り出し、VTOCをキャッシュメモリ上に読み出す。次にキャッシュメモリ上に読み出したVTOCを先頭から読み、対象ファイルを探し、その位置情報を読み出す(ステップ3004)。ステップ3005では対象となるファイルの位置情報からエクステント位置を調べ、各エクステントの先頭位置と、最終の位置を領域情報147に記録する。エクステントが複数ある場合には複数のエクステント情報をソートし、アドレスの小さい方から並べておく。ステップ3006でVTOCの対象ファイルの情報をディスク33に書き込む。

【0043】図15で移行処理を説明する。ステップ3101で、制御装置35内のコピーポインタ43をエクステント先頭アドレスに初期化する。ステップ3102でコマンドキュー内を参照し、ホスト1からのリード/ライト処理の要求が来ているかチェックする(ステップ3103)。処理要求がある場合には、キュー内の処理を実行する(ステップ3104)。ステップ3105では、旧記憶装置サブシステム2にコピーポインタ43の示す領域のリード要求を送る。

【0044】ステップ3106で旧記憶装置サブシステム2からリードデータを受け取り、ステップ3107で新記憶装置サブシステム3の当該アドレスにデータを書き出す。ステップ3108でコピーポインタ43に1加算する。ステップ3109でコピーポインタ43がエクステント最終アドレスを超えたかチェックし、まだ越えていない場合にはステップ3102に戻り、処理を繰り返す。最大ブロック数に達していれば、領域情報147を参照して次のエクステントがあるかチェックする(ステップ3110)。全てのエクステントについて処理が終わっていれば、移行処理は終了し、次のエクステントがあればコピーポインタを次のエクステント先頭アドレスに設定する(ステップ3111)。

【0045】次に図16で、移行処理中にホスト1からのリード要求が来た場合の新記憶装置サブシステム3での処理の流れについて説明する。まず、ホスト1から来たリード先アドレスと、新記憶装置サブシステム3で保持しているコピー対象ファイルのエクステントの情報とを比較していき、リード先アドレスが領域情報147のエクステント内に入っているかチェックする(ステップ

3201)。エクステント内に入っている場合には、次にコピーポインタ43とリード先アドレスとを比較する(ステップ3202)。コピーポインタ43がリード先アドレスよりも大きい場合には、新記憶装置サブシステム3には旧記憶装置サブシステム2からすでにデータがコピーされているので、新記憶装置サブシステム3からデータを読み出し(ステップ3203)、ホストにデータを送ることでリード処理を行うことができる。

【0046】ステップ3201エクステント内に入っていない場合は、ホストからのリード要求は今回の移行処理の対象ファイルに対するアクセスではないため、旧記憶装置サブシステム2からデータを読み出すことになる。旧記憶装置サブシステム2にリード要求を送り(ステップ3204)、旧記憶装置サブシステム2からリードデータを受け取る(ステップ3205)。またはステップ3202でコピーポインタ43がリード先アドレス以下と判定された場合にも、旧記憶装置サブシステム2からデータを読み出すことになる。最後にデータをホストに送ることで(ステップ3206)リード処理が完了する。

【0047】ここで、エクステントが多数存在する場合、ステップでのリード先アドレスがエクステント内にあるかのチェック処理のオーバーヘッドがかかる。そのため、リード処理の場合には、リード先アドレスの値に関わらず、無条件に旧記憶装置サブシステム2からの読み出しを実施してするという実装でも良い。

【0048】次に図17で、移行処理中にホスト1からのライト要求が来た場合の新記憶装置サブシステム3での処理の流れについて説明する。ステップ3301で、まず、ホスト1から来たライト先アドレスと、新記憶装置サブシステム3で保持しているコピー対象ファイルのエクステントの情報とを比較していき、ライト先アドレスが領域情報147のエクステント内に入っているかチェックする。エクステント内に入っている場合には、データ移行の対象となっているファイルに対する書き込みであるため、旧記憶装置サブシステム2に対してライト要求を出し、ホスト1から受け取ったライトデータを書き込む(ステップ3302)。次にコピーポインタ43とライト先アドレスとを比較する(ステップ3303)。コピーポインタ43がライト先アドレスより小さい場合には後で移行処理によって旧記憶装置サブシステム2から新記憶装置サブシステム3へとデータコピーが行われるため、今ディスク33に対してデータを書き込む必要はなく、新記憶装置サブシステム3に対しては何もせずにライト処理は終了する。そうでない場合、またはステップ3301でライト先アドレスが対象エクステント外と判定された場合には新記憶装置サブシステム3内のディスク33にデータを書き込む(ステップ3304)。

【0049】

【発明の効果】本発明の計算機システムでは、ホストコンピュータからの処理を受け付けながら旧記憶サブシステムから、新記憶サブシステムへのデータ移行が可能である。また、ファイル単位での移行が可能であり、移行したいデータだけを選択的に移行できる。また、移行中のデータは旧記憶装置サブシステムと新記憶装置サブシステムの2つに二重書きされるため、移行中にどちらかの記憶装置サブシステムに障害が発生しても移行処理を継続できる。

【図面の簡単な説明】

【図1】本発明を適用した計算機システムの構成例を示す。

【図2】本発明の計算機システムにおける、データ移行処理の流れを示す。

【図3】新記憶装置サブシステム3のデータ移行処理中のリード処理の流れを示す。

【図4】新記憶装置サブシステム3のデータ移行処理中のライト処理の流れを示す。

【図5】本発明を適用した第2の実施形態の計算機システムの構成例を示す。

【図6】本発明のコピーポインタとアクセスカウンタの管理方法を示す。

【図7】本発明の計算機システムにおける、データ移行処理の流れを示す。

【図8】本発明のデータ移行処理でコピーポインタを並べかえたときの状態を示す。

【図9】新記憶装置サブシステム3のデータ移行処理中のリード処理の流れを示す。

【図10】新記憶装置サブシステム3のデータ移行処理中のライト処理の流れを示す。

【図11】本発明を適用した第3の実施形態の計算機システムの構成例を示す。

【図12】第3の実施形態における、ファイルの管理方式を示す。

【図13】第3の実施形態における、領域情報の内容を示す。

【図14】本発明の第3の実施形態における、ファイル解析処理の流れを示す。

【図15】本発明の第3の実施形態における、データ移行処理の流れを示す。

【図16】本発明の第3の実施形態における、データ移行処理中のリード処理の流れを示す。

【図17】本発明の第3の実施形態における、データ移行処理中のライト処理の流れを示す。

【符号の説明】

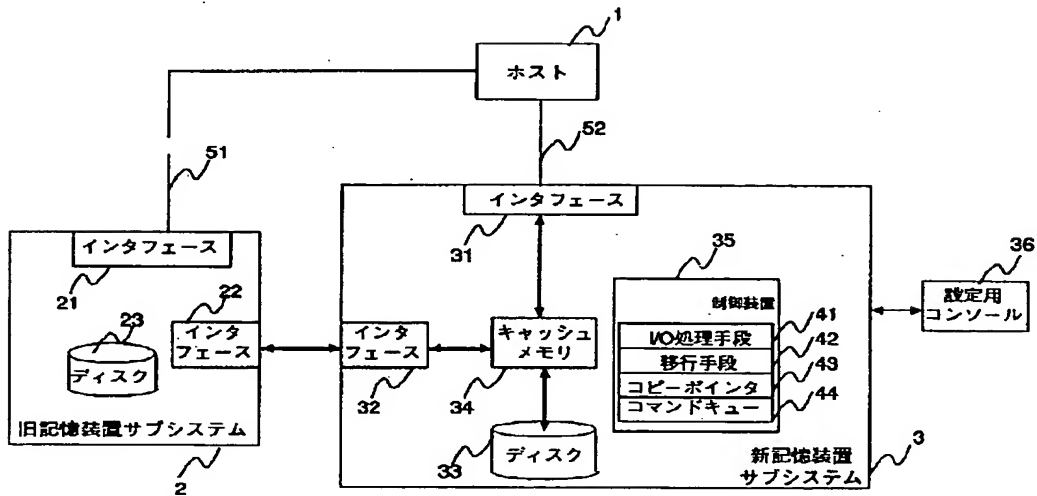
1…ホスト、2…旧記憶装置サブシステム、3…新記憶装置サブシステム、21…インタフェース、22…インタフェース、23…ディスク、31…インタフェース、32…インタフェース、33…ディスク、34…キャッシュメモリ、35…制御手段、36…

設定用コンソール、41…I/O 処理手段、42…移行手段、43…コピーポインタ、44…コマンドキュー、51…伝送路、52…伝送路、

143…コピーポインタ、145…アクセスカウンタ、147…領域情報。

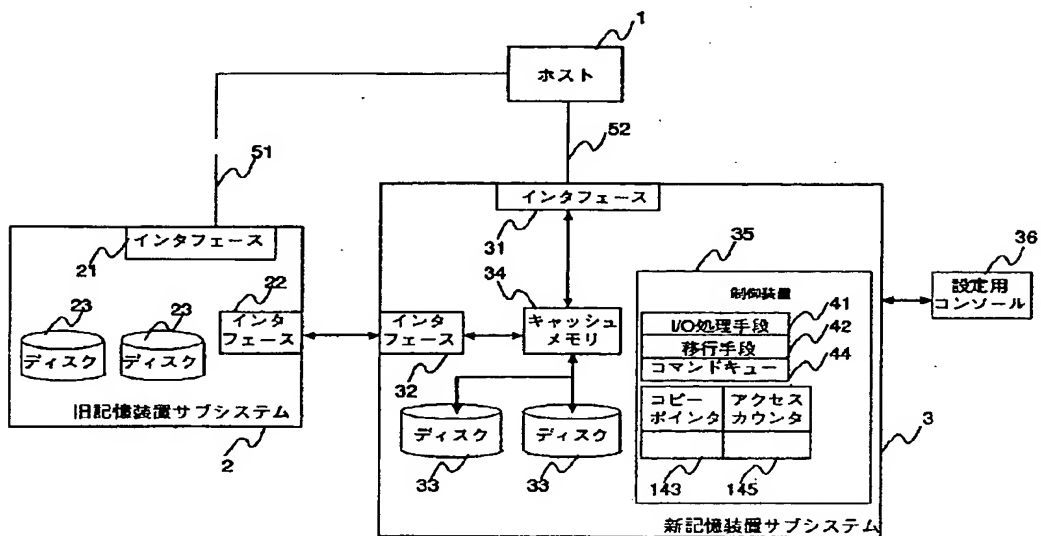
【図 1】

図 1



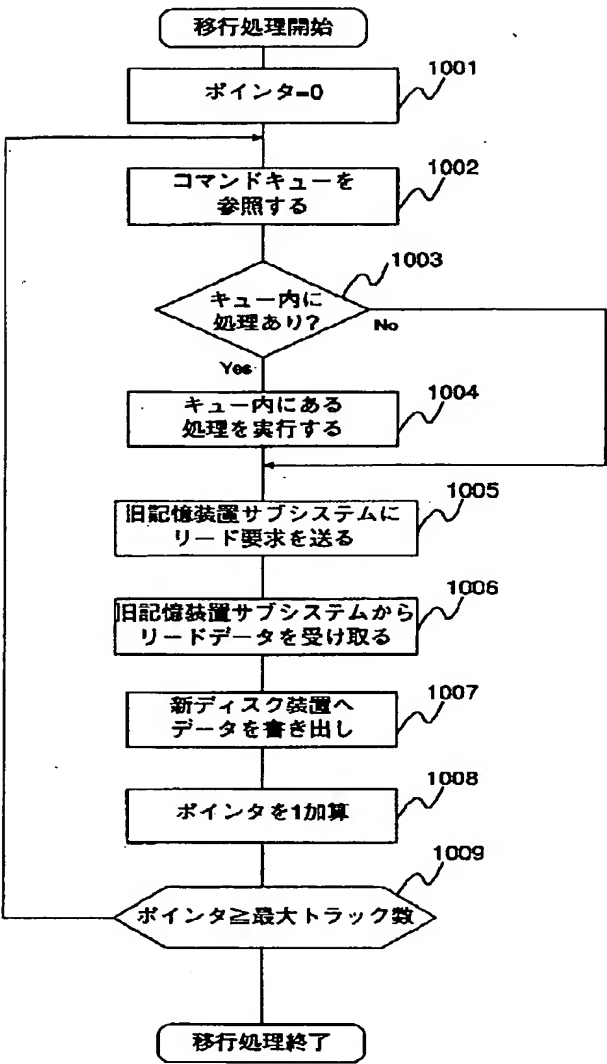
【図 5】

図 5



【図2】

図 2



【図13】

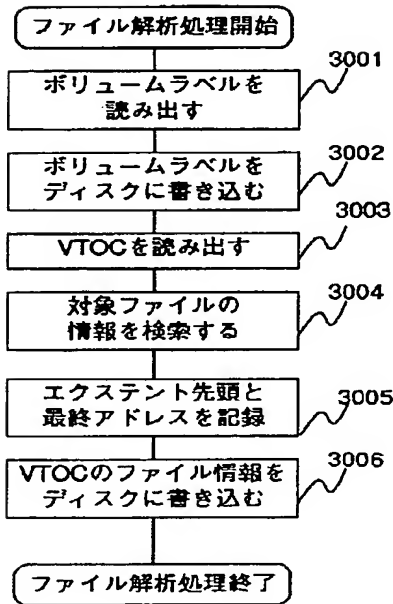
図 1 3

147

領域情報		
	先頭アドレス	最終アドレス
1	a1	a2
2	b1	b2
3	c1	c2
	⋮	⋮
n		

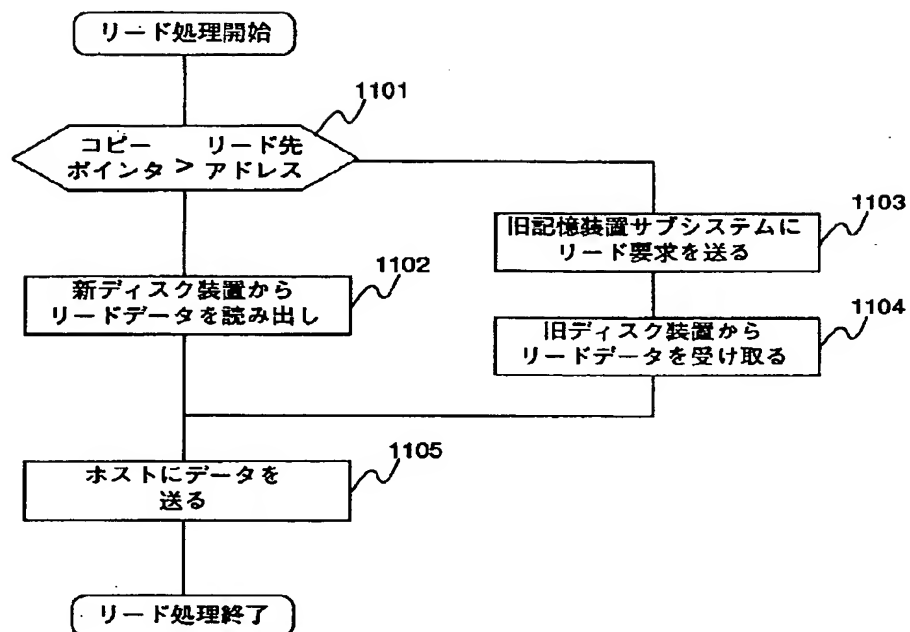
【図14】

図 1 4



【図 3】

図 3



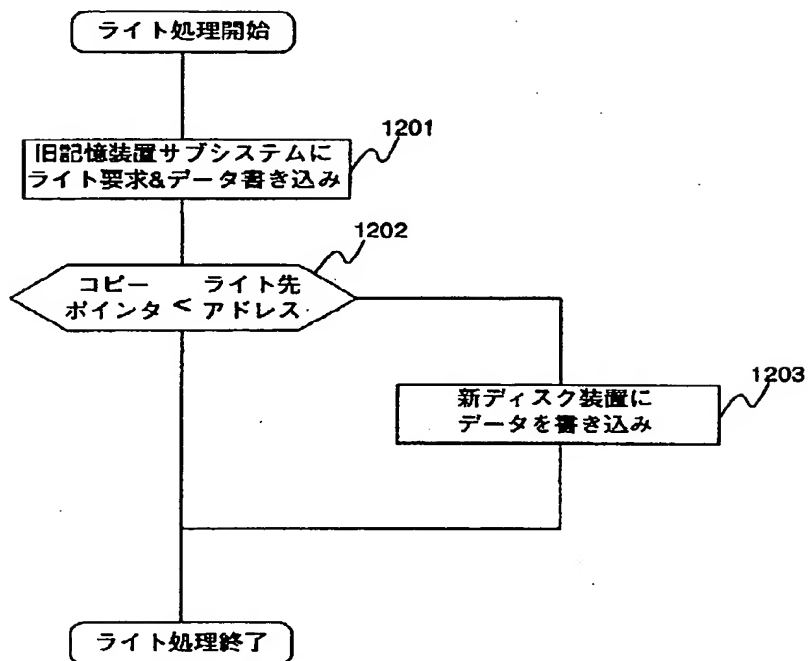
【図 6】

図 6

	先頭アドレス	最終アドレス	コピーポインタ	アクセスカウンタ
1	0	a-1	0	0
2	a	2a-1	a	2
3	2a	3a-1	2a	1
	⋮	⋮		
n				

【図 4】

図 4



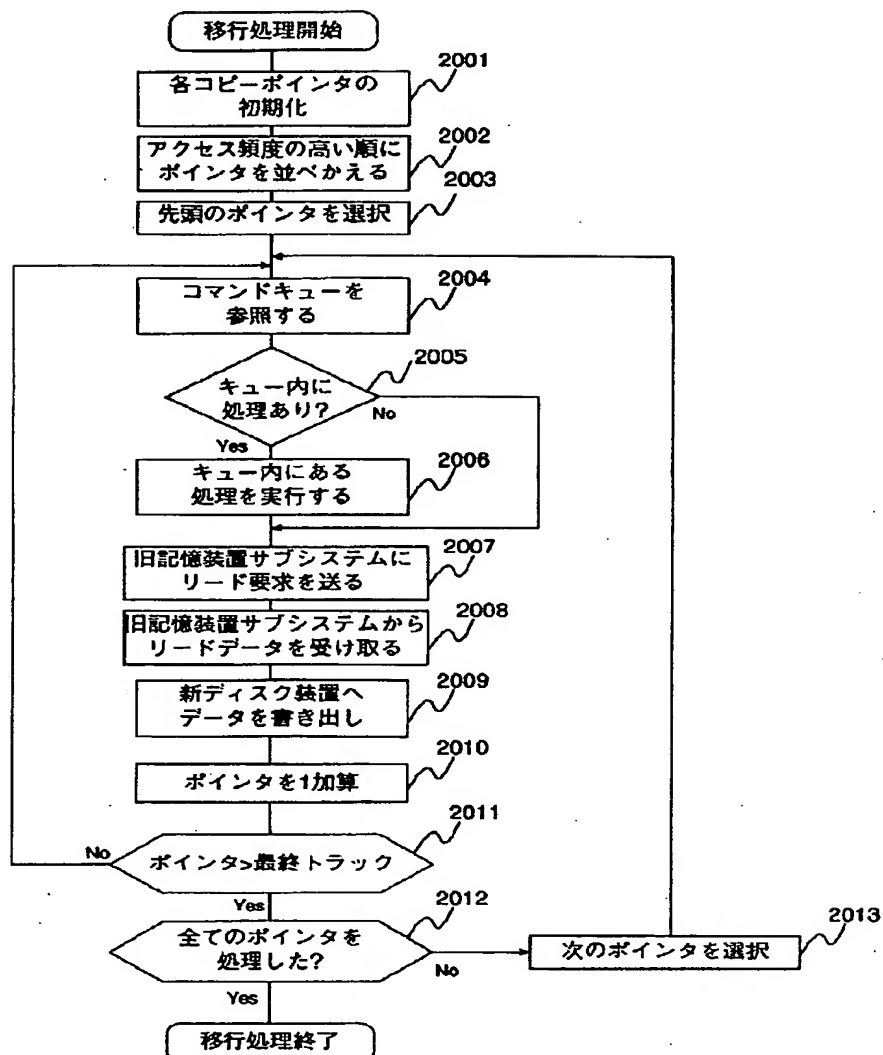
【図 8】

図 8

	先頭アドレス	最終アドレス	コピーポインタ	アクセスカウンタ
1	a	2a-1	a	2
2	2a	3a-1	2a	1
3	0	a-1	0	0

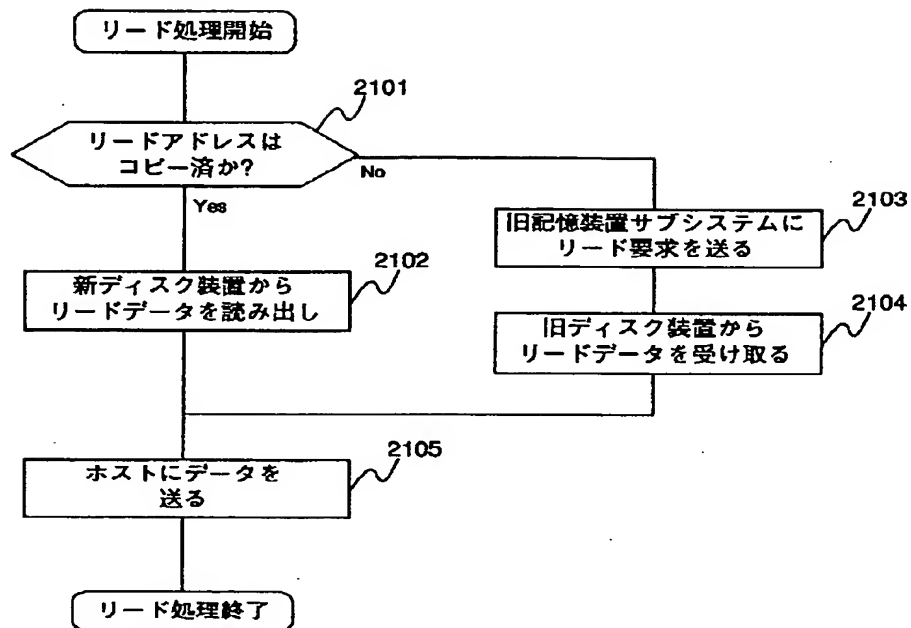
【図7】

図7



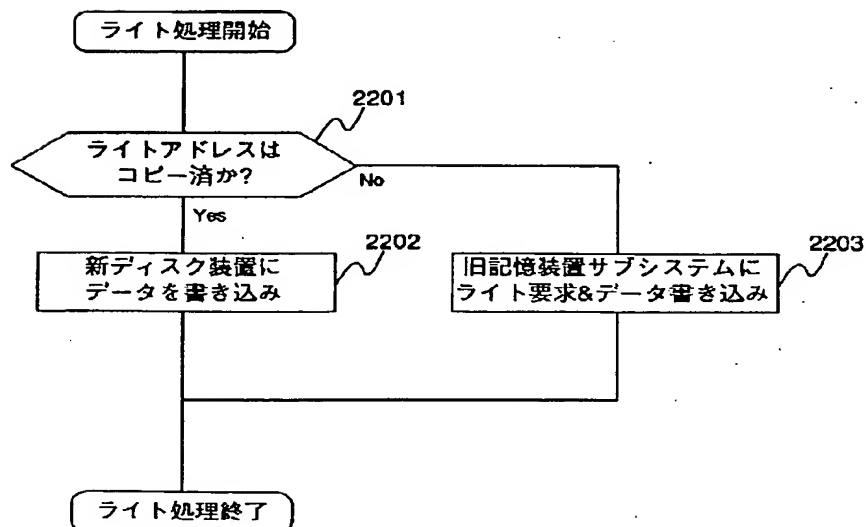
【図9】

図 9



【図10】

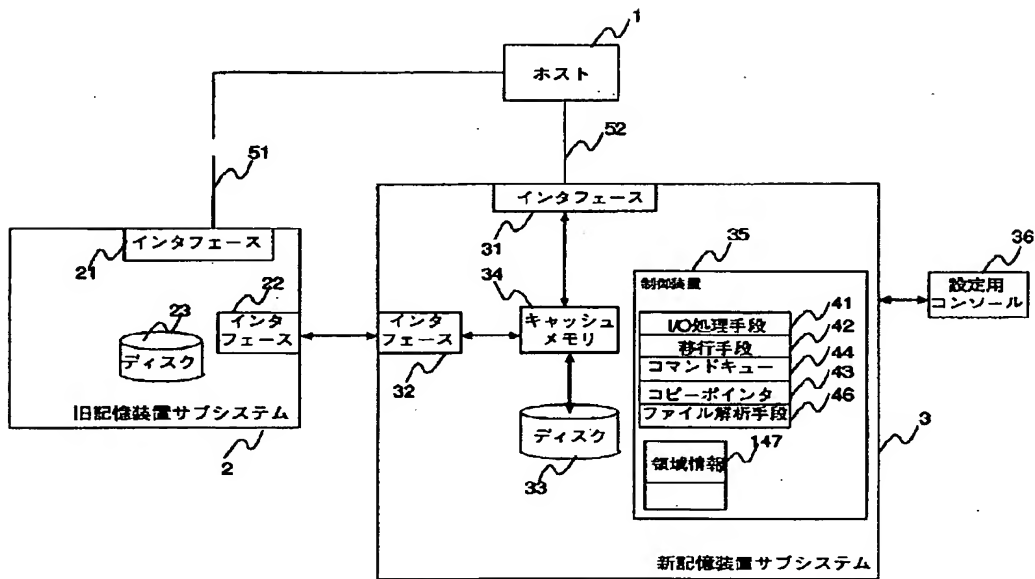
図 10





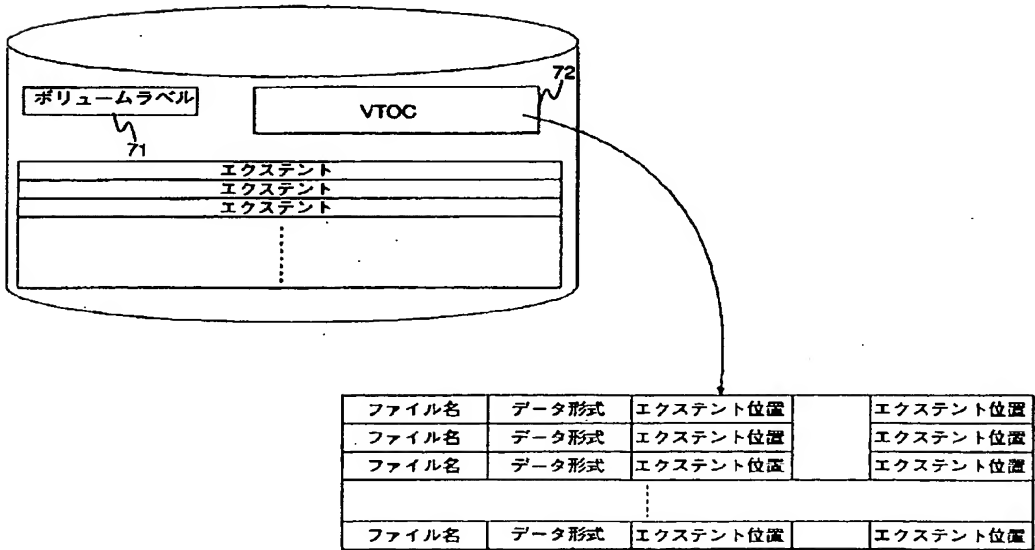
【図11】

図 1 1



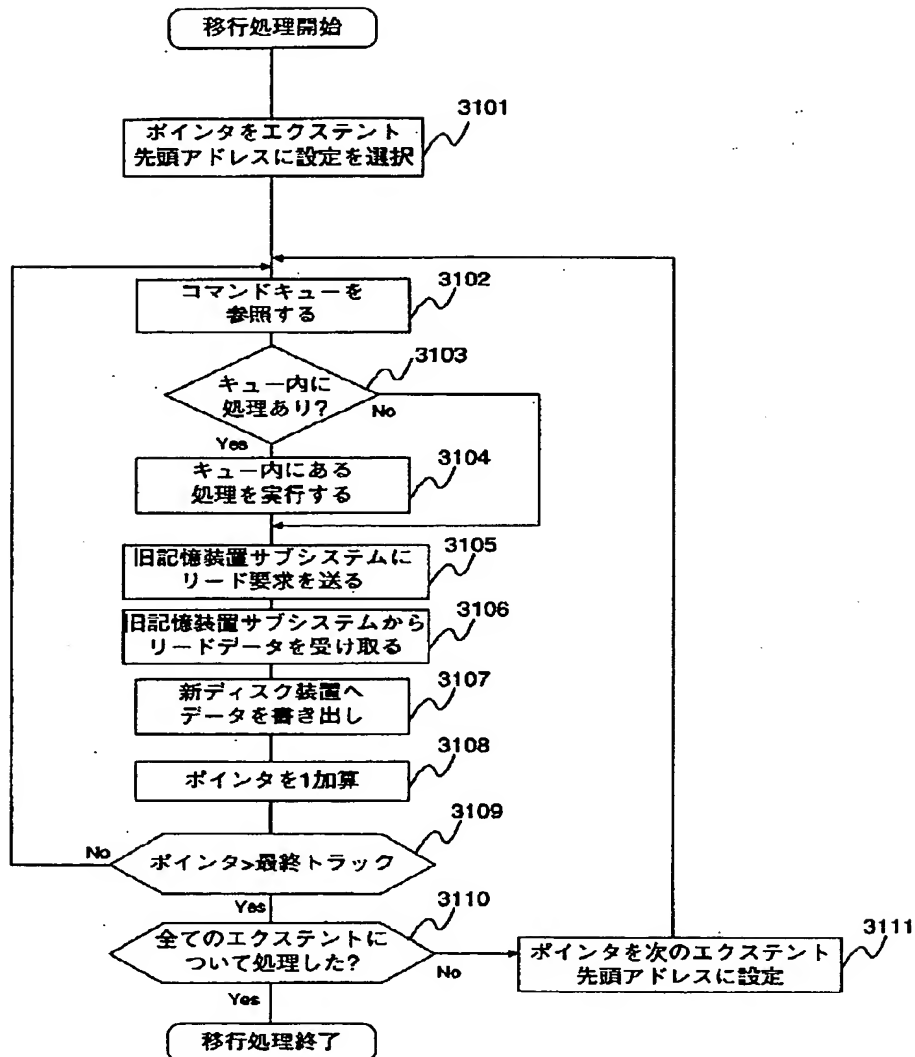
【図12】

図 1 2



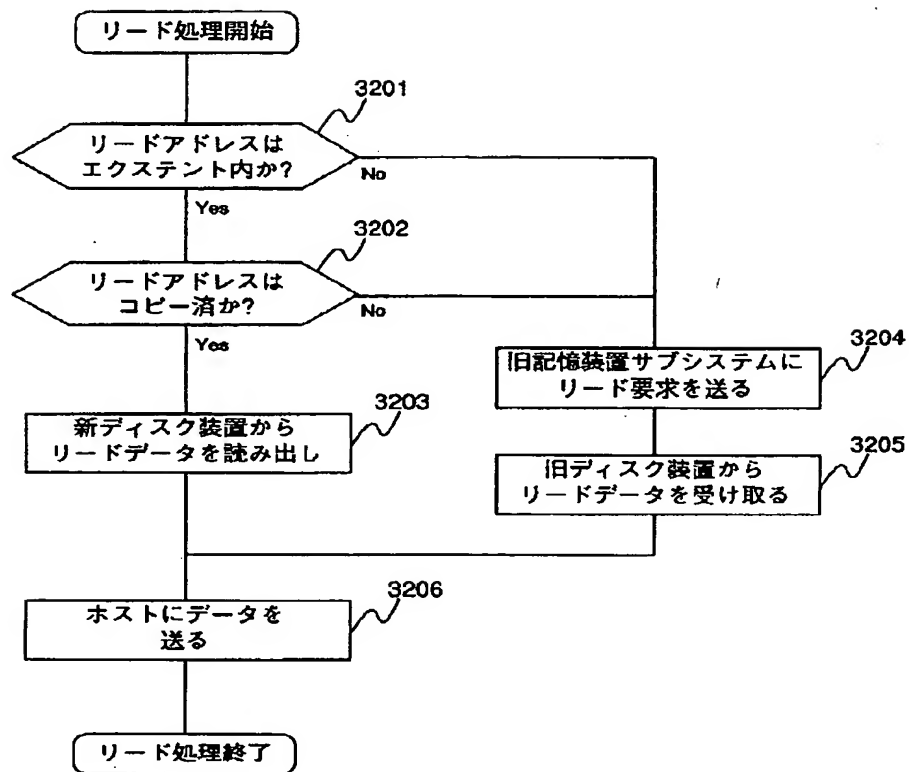
【図15】

図15



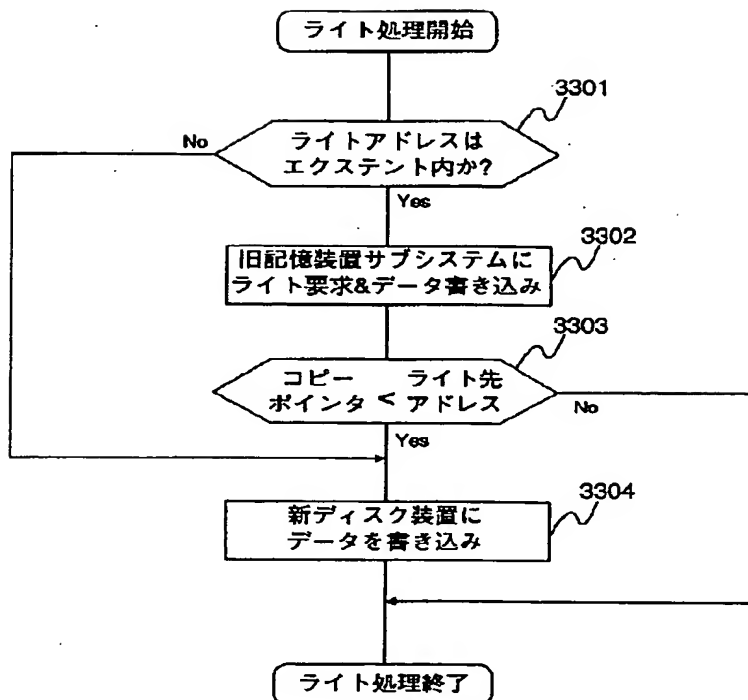
【図16】

図16



【図17】

図17



フロントページの続き

(72)発明者 倉野 昭

神奈川県小田原市国府津2880番地 株式会  
社日立製作所ストレージシステム事業部内

(72)発明者 神林 公咲

神奈川県小田原市国府津2880番地 株式会  
社日立製作所ストレージシステム事業部内

Fターム(参考) 5B065 BA01 CE11 CE21 EA31

5B082 AA00 CA10 DE04 DE06 HA05